Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

1990-03

Measure of effectiveness for amphibious ship loading

Schneider, Joseph M.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/30717

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun

http://www.nps.edu/library

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

AD-A230 307

NAVAL POSTGRADUATE SCHOOL
Monterey, California

DTIC
ELECTE
JAN 0 3 1991
S D
D

THESIS

MEASURE OF EFFECTIVENESS FOR AMPHIBIOUS SHIP LOADING

by

Joseph M. Schneider

March 1990

Thesis Advisor: Gordon H. Bradley

Approved for public release; distribution is unlimited

91 1 2 058

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704 0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
MEASURE OF EFFECTIVENESS FOR AMPHIBIOUS SHIP LOADING

12 PERSONAL AUTHOR(S)
Schneider, Joseph M.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	1990, March	84

16 SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Amphibious Ship Loading; Embarkation Planning

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

A critical factor in the success of an amphibious operation is how well the load plan supports the landing plan. The current manual system for ship load planning is time consuming and subject to error. A computer system currently under development by a contractor will decrease planning time and reduce mistakes by automating many details of the planning process. A method to assess the quality of load plans and make comparisons among them is also essential to improved planning. The scoring algorithm developed in this paper implements a measure of effectiveness (MOE) to make these comparisons by scoring a load plan's ability to support the landing plan. The algorithm provides the ability to differentiate qualitatively among loads by computing penalty scores for the critical areas of equipment left behind, compartment location, and compartment

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL Prof. Gordon H. Bradley	22b TELEPHONE (Include Area Code) (408) 646-2359 22c OFFICE SYMBOL Code 55Bz

DD Form 1473, JUN 86 Previous editions are obsolete SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603

UNCLASSIFIED

#19 - ABSTRACT - (CONTINUED)

access. The trade-off of lightly loading the ship for flexibility versus leaving critical cargo behind is implicitly considered. Raw and normalized scores in each area and a total score are provided to the user. The MOE produced by this scoring algorithm is cost effective, easy to implement, easy to use and, if fully developed and adopted, will lead to improved loading of amphibious ships.

Approved for public release; distribution is unlimited

Measure of Effectiveness for Amphibious Ship Loading

by

Joseph M. Schneider
Lieutenant Commander, United States Navy
S.B., Massachusetts Institute of Technology, 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1990

Author: [redacted]
Joseph M. Schneider

Approved by: [redacted]
Gordon H. Bradley, Thesis Advisor

[redacted]
William J. Walsh, Second Reader

[redacted]
Peter Purdue, Chairman
Department of Operations Research

ABSTRACT

A critical factor in the success of an amphibious operation is how well the load plan supports the landing plan. The current manual system for ship loading planning is time consuming and subject to error. A computer system currently under development by a contractor will decrease planning time and reduce mistakes by automating many details of the planning process. A method to assess the quality of load plans and make comparisons among them is also essential to improved planning. The scoring algorithm developed in this paper implements a measure of effectiveness (MOE) to make these comparisons by scoring a load plan's ability to support the landing plan. The algorithm provides the ability to differentiate qualitatively among loads by computing penalty scores for the critical areas of equipment left behind, compartment location, and compartment access. The trade-off of lightly loading the ship for flexibility versus leaving critical cargo behind is implicitly considered. Raw and normalized scores in each area and a total score are provided to the user. The MOE produced by this scoring algorithm is cost effective, easy to implement, easy to use and, if fully developed and adopted, will lead to improved loading of amphibious ships.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

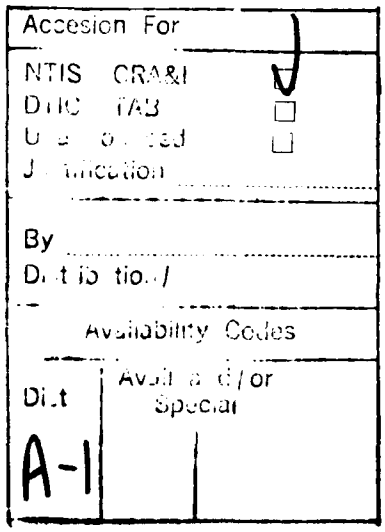


TABLE OF CONTENTS

I. INTRODUCTION ---------- 1

A. INTRODUCTION ---------- 1

B. APPROACH TO PROBLEM ---------- 2

C. OVERVIEW ---------- 4

II. BACKGROUND ---------- 7

A. DESCRIPTION OF PROBLEM ---------- 7

B. DIRECTIONS FOR SOLUTIONS ---------- 12

III. COMPUTER AIDED EMBARKATION MANAGEMENT SYSTEM (CAEMS) ---------- 14

A. HARDWARE REQUIREMENTS ---------- 14

B. SOFTWARE REQUIREMENTS ---------- 15

C. THE SYSTEM ---------- 15

D. MAJOR FEATURES ---------- 16

E. TYPICAL PLANNING SESSION FOR TEO ---------- 17

F. THE PRORATION PROCESS ---------- 19

IV. MEASURE OF EFFECTIVENESS FOR LOAD PLANNING ------ 22

A. THE LOADING PROBLEM ---------- 22

B. APPROACHES ---------- 23

C. OPTIMIZATION MODELS ---------- 24

D. SCORING ALGORITHM APPROACH ---------- 29

V. PROPOSED APPROACH ---------- 31

A. MEASURE OF EFFECTIVENESS (MOE) REQUIREMENTS - 31

B. SCORING ALGORITHM ---------- 33

VI. IMPLEMENTATION/VALIDATION ---------- 42

A. IMPLEMENTATION ---------- 42

B. VALIDATION ---------- 46

VII. CONCLUSIONS AND RECOMMENDATIONS ---------- 48

APPENDIX A: PAL CODE AND SAMPLE TABLES ---------- 50

APPENDIX B: GLOSSERY OF TERMS ---------- 66

LIST OF REFERENCES ---------- 69

BIBLIOGRAPHY ---------- 70

INITIAL DISTRIBUTION LIST ---------- 72

ACKNOWLEDGMENTS

I would like to thank the many individuals and organizations who assisted me in my research for this thesis. Captain Santamaria, USN, Commander, Amphibious Squadron 3, and his staff helped me to identify an area for research during my experience tour. Frank discussions with him and with his staff were invaluable. Captain Dean Fish, USMC, and others from LOG AIS Section, Logistics Information Systems Branch, Logistics Plans, Policies and Strategic Mobility Division, Installations and Logistics Department, Headquarters Marine Corps, provided the source code and documentation for the CAEMS-1 prototype. They provided coordination for my attendance at the Senior Marine Corps Logisticians' Conference where I observed demonstrations of CAEMS and other automated logistic systems. Michael Innella and William Karlson from Stanley Associates were most generous with their time and explanations of the CAEMS-1 prototype at these demonstrations. Many individuals from the operational side of the Marine Corps aided my efforts as well.

My special thanks to Captain Rick Reece, USMC, for his answers to my many questions about ground combat. Captain Ray St-Germaine, USMC provided me with insight into the origins of the CAEMS system as well as information about

load planning in general. Captain Jim Shulson, USMC, from the Landing Force Training Command, Pacific provided numerous publications and handouts from the load planning course of instruction during my recent visit to his office. His assistance led to a better understanding of the integration of the load plan with the landing plan.

Finally I would like to thank my wife and children. I spent many hours traveling to various commands for information, and uncountable hours at my computer learning Paradox, C, and AutoCAD and studying CAEMS-1. Their patience and understanding made this work possible.

I. INTRODUCTION

A. INTRODUCTION

Constructing amphibious ship load plans that support the landing plan is critical to a successful amphibious assault. The current manual planning system is tedious, time consuming and subject to error. A computerized approach to the problem is needed to improve load planning accuracy and decrease required planning time. In addition, a way to measure the quality of a load plan would improve the planner's ability to choose the best plan among several alternatives. This thesis provides a scoring algorithm that provides a measure of effectiveness (MOE) to accomplish this.

Penalty scores are developed with reference to a landing plan in order to score the load plan's ability to support the landing. The algorithm computes scores in three key areas:

1. The amount of cargo that must be left behind due to lack of space.
2. The difficulty of off-loading cargo from each compartment of the ship.
3. The amount of free space, by percentage, when cargo is off-loaded from a compartment in the order specified by the landing plan.

The free space score is a surrogate for flexibility in the off-load. The more free area available, the easier it is

to off-load cargo in the required order quickly. Inherent in these penalties is the conflict between lightly loading the ship for flexibility versus loading the maximum amount of cargo The final output provided to the planner is the three raw scores, three normalized scores and an overall score. These scores enable the planner to improve his decision making and will lead to better load planning.

The focus of this work is on the individual ship and the people who must develop load plans for embarked units to carry out a mission. The scope is restricted to the loading and off-loading of cargo, and does not examine the embarkation of personnel. The concepts explored could, with some modification, be used at the amphibious squadron level as well.

B. APPROACH TO PROBLEM

A computerized approach was taken for several reasons. A human planner currently must keep track of many details when planning a load. There are hundreds of different cargo items each with different heights, widths, lengths and weights. There are dozens of compartments on a ship where cargo can be loaded. There are also constraints on where cargo can be stored. In addition, the planner must keep track of which unit owns the cargo and when it is to be off-loaded during an amphibious landing. The sheer volume of data that must be organized leads logically to a database approach to reduce the burden on the planner. The computer

reduces error by providing consistency and accuracy for these details. Loads that are infeasible because they violate constraints can be weeded out so that the planner considers only plans that are physically possible. A computerized system under development by a contractor, the Computer Aided Embarkation Management System (CAEMS), incorporates these features.

The CAEMS database was the starting point for the scoring algorithm developed here. The ability to plan a load quickly and score it on the computer allows users to create alternate plans and play "what-if" scenarios to compare the values of the MOE and choose the best plan.

In implementing a MOE, several considerations are important. The amount of additional data that must be collected should be kept to a reasonable level. A MOE should be complete enough to capture the salient features of the real world but should not be so detailed that the computations cannot be efficiently performed. The conclusions drawn must be supported by the input data available. A balance must be reached with the needs of the users as the driving factor. The approach taken must be understood by the user or it will not be effectively utilized. The outputs must aid in the decision making process. In addition, the implementation costs must not be greater than the benefits provided. The MOE implemented by the scoring algorithm developed in this paper adheres to

these principles. If fully developed and then adopted, it can play a major role in improving amphibious ship load planning.

C. OVERVIEW

Chapter II is a discussion of the problem background. The conflict between loading the maximum amount of cargo versus the need for flexibility during the landing phase of the operation is discussed. The key shipboard players are identified. The inputs to the planners are reviewed and the essential outputs of the planning process are identified.

Chapter III reviews an existing prototype, the Computer Aided Embarkation Management System (CAEMS) [Refs. 1, 2]. This is an ongoing project that should be available to the fleet in the near future. It uses primarily off-the-shelf software to develop a relational database for the embarking units and their equipment and a computer representation of the characteristics of a particular ship. The logical presentation of large amounts of data makes this system particularly valuable to the planner. Some key features and error checking capabilities of CAEMS are discussed. One major shortcoming of this system is the inability to distinguish good load plans from inferior ones. Addressing this issue is critical to improving the performance of planners.

Chapter IV addresses the issue of choosing a mathematical model that would provide an optimal or "near"

optimal solution. The nature of the problem is examined and similarity to the classical "knapsack" model is discussed. Next the suitability of a multiple objective model is examined. A goal programming approach to the model is also discussed. For various reasons related to the amount of user data required, numerical size of the problem and inability to obtain required model inputs, these models were rejected. Instead, the concept of using a scoring algorithm to differentiate among loads was developed. The key features of such a system are identified.

Chapter V develops the scoring algorithm. It depends on what equipment is loaded, where it is loaded, and how much free area remains in each ship. The details of the model, including the algorithm, user input requirements, and output values are provided. These outputs include both raw and normalized scores so that a planner can qualitatively compare one load plan against another and can determine the area where differences occur.

Chapter VI provides the details to implement the scoring algorithm based on the database provided by CAEMS. By building on this existing system, the algorithm can be implemented efficiently with a minimum of additional programming. Data input from the user is kept to a minimum. The additional data elements and database tables are identified and a scheme for validating and adjusting the various scoring penalties is provided.

Chapter VII summarizes the critical issues in load planning and presents conclusions. The contribution of the MOE developed is emphasized. Follow-on research could develop "typical" missions and landing plans. Running the scoring algorithm against these plans could lead to refined penalty rates and weighting factors.

II. BACKGROUND

A. DESCRIPTION OF PROBLEM

This paper describes and analyzes the problem of loading amphibious ships from the limited viewpoint of a naval planner supporting the objectives of an amphibious operations. The goal is to load the ships of an amphibious task group in the "best" possible manner. Many issues that are pertinent to the conduct of warfare that do not have an immediate impact on the decisions of a planner have, therefore, been excluded from this study. With this view in mind, the following is a brief description of amphibious operations as it pertains to this topic. The discussion that follows is paraphrased from Ground Combat Operations [Ref. 3].

An amphibious operation is an attack launched from the sea by naval and landing forces, embarked in ships or craft involving a landing on a hostile shore. [Ref. 4] Any amphibious operation is complex and requires detailed planning and coordination. Amphibious operations are conducted for the following reasons:

1. Obtain a lodgement in order to pursue further combat ashore.
2. Obtain sites for advanced naval or air bases.
3. Deny the enemy the use of the area.

The primary type of amphibious operation is the amphibious assault which involves establishing a force on a hostile shore. [Ref. 4] The main feature of the assault is the need to build combat power ashore from a zero base line as quickly as possible. Other types of operations include the raid, demonstration, and amphibious withdrawal. These share many characteristics of the assault although on a more limited scale. The primary difference is that they do not involve the permanent establishment of military forces ashore.

The principal planners for an amphibious operation consist of both Navy and Marine Corps personnel. On the Navy side the squadron commander will be in charge of an amphibious ready group consisting of several amphibious ships. The squadron commander becomes the Commander Amphibious Task Force (CATF). On the Marine side, a senior colonel is designated the Commander Landing Force (CLF). Various ground and air element personnel also take part in the planning as do individuals from each ship in the squadron.

A key area in this planning process is deciding what equipment to load onto the amphibious ships and how to load it. This is a combined responsibility of the marine corps and shipboard personnel. There are two conflicting goals in this loading process. The first is to load as much equipment and ammunition as is physically possible cramming

cargo into every "nook and cranny." This type of loading is known as an "administrative" loading. A "combat" loading, on the other hand, has as the primary objective ensuring that equipment is loaded so as to be immediately available in the order it needs to reach the beach.

The entire planning process of amphibious ship loading is directed toward supporting the landing plan (the entire process and detailed instructions for the ship-to-shore movement). The landing plan in turn must support the concept of operations and the scheme of maneuver ashore. This plan must provide for maximum shock effect, depth to the assault, and a rapid buildup of forces ashore. Maximum use of helicopter, amphibious assault vehicles (AAVs) and landing craft is critical. Flexibility to respond to changing situations and to exploit enemy weakness is also critical. Therefore, the assault and initial unloading period must be tactical in nature and must be responsive to the landing force requirements ashore. This phase of the operation consists of various units scheduled to go ashore at various times. Groups of men and equipment on an individual ship are organized into serials. Each piece of cargo on a ship is assigned as part of a serial for identification. These serials are called out for movement ashore at a given time, for scheduled waves, or as dictated by requirements ashore for on-call waves. Scheduled waves transport the initial assault elements either by air or sea.

On-call waves of men and equipment are subject to immediate call with their need anticipated at an early hour in the assault, however the exact timing cannot be determined in advance. They are normally composed of reserves, direct support artillery, combat engineers, tanks, light armor, and landing support elements. This requirement for quickly supporting a changing situation ashore means that combat loading is critical to successful amphibious operations.

Consideration in the on-load must be given to order of off-load and flexibility. As a result, a ship cannot be loaded to its "theoretical" maximum but must maintain open space and aisles to ensure this flexibility. Thus a natural conflict is created with trade-offs between the amount of material that can be carried and the ability to support the operation ashore in the required sequence. This problem is exacerbated by the fact that many different units are involved with various equipments that, in most cases, exceeds the capacity of the ships involved. The CLF and the CATF must, therefore, make critical decisions on what to leave behind and how to load the equipment that is carried based on the anticipated mission.

The planning process for an amphibious operation takes place over a period of several weeks to several months. Marine Corps units are assigned, individual ships of the task group are determined, and augmentation teams, such as Explosive Ordinance Disposal (EOD) teams, surgical units,

SEAL teams, etc., are added to the units involved. These units and detachments have organic equipment and supplies that must be loaded onboard the ships of the task force. The quantity of equipment available as well as the amount authorized plays a role in the determination of the total load. In addition, a standard load of ammunition known as LFORM is carried by each ship according to class of ship. This load is also dependent on the availability of ammunition and varies from deployment to deployment. The entire required load is known to the navy planner at least several weeks prior to the actual onload of equipment. Material is loaded at different times for different units, however, and some rearrangement might be necessary as additional unit equipment is loaded. For the most part actual cargo to be loaded is known well in advance of departure date, therefore the planner has the time to carefully consider alternate load plans and develop one that meets the tactical requirements of the CATF.

One of the hardest, and most important decisions, is what to leave behind when required load exceeds the available space, as is usually the case. This problem is accentuated by the need to leave some amount of free space and aisles to allow for flexibility and ensure a combat loading where equipment can be sent ashore in the required order. The planner must make three decisions: what equipment to load on each ship, in what location and in what

order. As a consequence of these decisions what equipment is left on the pier is also determined.

The planner must produce several reports as a result of his decisions. The reports describe which units and equipment are loaded on each ship and include a ship's cargo manifest for each ship. Additionally, a template of each stowage area of the ship and the location of cargo in each of these spaces is produced.

B. DIRECTIONS FOR SOLUTIONS

Currently the complex problem of what to load and where to load is done entirely by hand. There is no consistent criteria to determine what a "good" load is versus a "bad" load. The planner is faced with massive amounts of data that must be massaged by hand. Decisions are made based on experience with questions such as "how did we load it last time?", which becomes the main driver for the current plan. Several possibilities exist to improve on the system. In general terms, there are multiple criteria to consider, such as taking as much as possible, versus the need for flexibility during off-load. The sheer amount of data lends itself to some sort of automation process to relieve the planner of a large part of the problem: keeping straight all the various units and their equipment. The use of a computer system to organize this information and to automate the planning process to some extent would be extremely beneficial. In addition, if such a system could help a

planner to distinguish between "good" and "bad" loads, improvements could be made in the way that amphibious ships are loaded. One clear and immediate advantage of such a system would be consistency. Since the data and the load plan would be available on computer the same plan could be used, after modification for changes in units or equipment, by future planners. Improved solutions would be possible over time as planners learned from past mistakes and these improvements recorded by the computer system.

III. COMPUTER AIDED EMBARKATION MANAGEMENT SYSTEM (CAEMS)

A computer aid for the loader does exist in the prototype stage of development. It is fully described in [Ref. 1] and [Ref. 2]. CAEMS is an ongoing contractor effort under Headquarters Marine Corps. The prototype ship modelled was the LHA-5. Currently, funding has been provided to develop the system for other amphibious platforms for introduction into the fleet. This effort has gone a long way toward easing the burden of load planners. The major objectives are:

1. To provide an interactive computer tool to assist in embarkation planning and execution.
2. To reduce the time required for planning and execution.
3. To provide the ability to respond rapidly to changes in shipping availability/mix and/or equipment configuration/density changes.
4. To provide a database for embarkation reports and information about embarked equipment and supplies.
5. To provide ship loading plans.
6. To provide trim, stability, and stress information (not implemented in the prototype, although provisions have been made to incorporate this feature in a future version).

A. HARDWARE REQUIREMENTS

The hardware requirement for the prototype is an IBM-AT compatible microcomputer, with Enhanced Graphics Adaptor (EGA) and monitor. One high-density 5 1/4" floppy drive, a

20 Megabyte hard drive, 640k of RAM, and a math coprocessor are needed. Due to the intensive database and CAD functions performed by the system, an 80386 based system and Video Graphics Array (VGA) are highly desirable to speed up operations and increase template resolution. A graphics capable printer is required for output and a plotter is highly desirable. In addition to required memory, a three megabyte RAM disk would reduce execution time.

B. SOFTWARE REQUIREMENTS

CAEMS was developed using commercial off-the-shelf software to speed development time and take advantage of excellent packages already in existence. Therefore, in order to use the CAEMS prototype the following software is also needed:

1. AutoCAD release 10, by Autodesk (a computer-aided design program).
2. Paradox version 3.0, by Borland International (a relational database).
3. The runtime version of Paradox 3.0 may be used in place of the full software package.
4. Micosoft DOS version 3.1 or above (the microcomputer operating system).

C. THE SYSTEM

CAEMS consists of a specialized application of the Paradox Data base written in Paradox Application Language (PAL), special interface and mathematical routines written in Microsoft C, and the templating subsystem using AutoCAD.

The result is a user-friendly, menu-driven interface that is easy to learn and use. The emphasis on software development was to provide a full set of tools to the Team Embarkation Officer (TEO) for quickly preparing a detailed load plan for a ship with required reports and diagrams. There are several different components of the CAEMS database, which are maintained by users other than the TEO. Ship reference data, such as compartments, zone constraints, cargo flow paths and digitized ship drawings are maintained in the ship reference portion. U.S. Coast Guard stowage compatibility groupings, supply codes, etc., are maintained in the CAEMS reference directory. This information is used as the embarkation planner prepares new load plans. The embarkation planner can view and edit this general reference data as required for the particular exercise or mission being planned. In addition, he aggregates information from deploying units or detachments, such as that provided by Standard Embarkation Management System (SEMS). The last step is for the unit planner to assign units to available shipping, so that all cargo for a given ship is labelled and made ready for import by the individual team embarkation officer.

D. MAJOR FEATURES

Some important features of the system include:

1. Data import allows SEMS data to be translated, normalized and converted to the CAEMS database tabl

structure. Consistency checks are performed and questionable data is highlighted.

2. Data base View/Edit is a menu driven means to review and edit data as required.

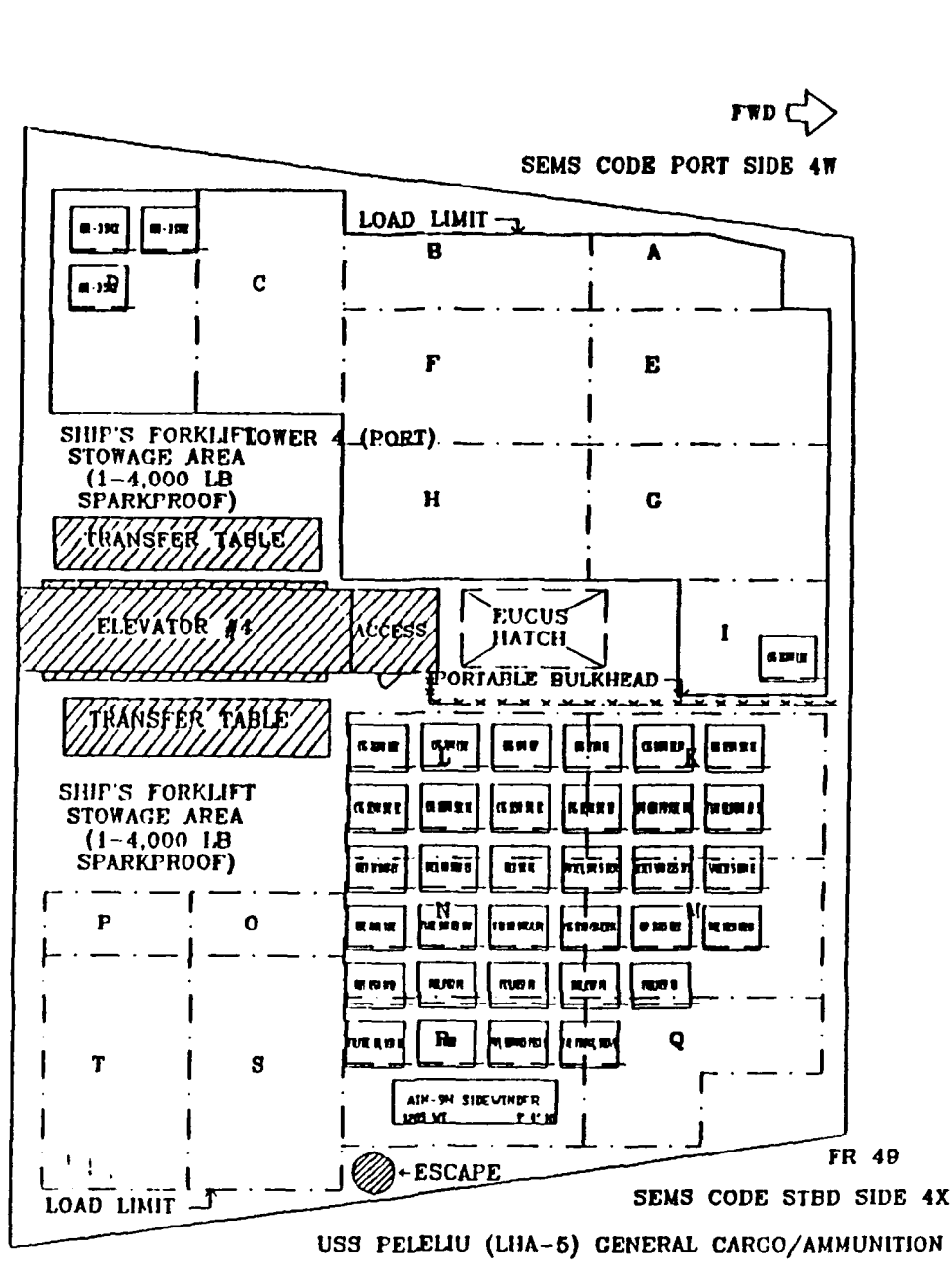
3. User-defined queries are a means for adhoc reports to be generated from the database.

4. Data consistency and validation is a major part of the system at every level.

5. Cargo templating is a computerized means to generate and position standard vehicle and pallet templates. The ability to produce detailed templates is also provided. As cargo is placed, checks are made against placement constraints and compatibility constraints with errors flagged for user review. Figure 1 is a typical template printout of a ship compartment.

E. TYPICAL PLANNING SESSION FOR TEO

The necessary data are imported from SEMS system or is entered by hand. The TEO can select a specific plan to edit or create a new plan. He initializes the cargo load and manually "seeds" the cargo for automatic proration (the process of assigning cargo to individual compartments and zones onboard the ship). The entire cargo list can be manually prorated if desired. Data tables are checked for inconsistencies and unit and cargo information updated as necessary. Violation checks are performed to ensure all constraints have been satisfied and all stowage rules followed during manual proration and templating routines. The TEO then selects cargo from individual compartments and using AutoCAD performs the placement operation where cargo is placed in an exact location in each space. Once this placement function is complete, error checking is again



USS PELELIU (LHA-5) GENERAL CARGO/AMMUNITION

SCALE: 1/8" = 1'

LOWER 4 COMPARTMENT 6-49-0-M

SCALE (FEET)

0 5 10 15 20 25

AVAILABLE SQ FT: 1745*
AVAILABLE CU FT: 15705*
PALLET SQUARES: 131*

ELEVATOR CAPACITY: L 16'2" X W 4'4" X H 5'5" MAX WT 12,000 LBS
L 5.541M X W 1.299M X H 1.701M

EUCUS HATCH SIZE: L 9'10" X W 5'6"
L 2.949M X W 1.65M

FIRE FIGHTING DEVICES: SPRINKLERS

RESTRICTIONS

MINIMUM CLEARANCE 9'0" 2.7M

MAXIMUM ALLOWABLE STOWAGE HEIGHT 9'0" 2.7M

Figure 1. CAEMS Deck Drawing

performed to check for constraint violations during placement. The planner is then ready to produce plots of the load plan and numerous embarkation reports. At any time in this process the database tools of the system can be used to view and edit any desired table. Specialized queries and reports can be produced as desired. The last step in the process is a hard disk cleanup to reduce the storage required for the database and a backup to a floppy disk to protect the data.

F. THE PRORATION PROCESS

The prototype automatic proration process is of major interest for this thesis. It is here that the opportunity exists to use a "smart" algorithm to help the TEO to produce a feasible load. In particular, the prototype flags mistakes such as Coast Guard Class incompatibility, violation of height and weight constraints, violations of no stow zones and ensures that no cargo is stored in a location that it is physically impossible for it to reach. The automatic proration process is a routine written in C that checks if cargo can be placed in a particular location. First all priority cargo is placed, based on the priority number. These numbers are a simple one to whatever number desired by the planner and are a strict ranking, not a grouping of priorities. The routine checks to see if a valid on-load path exists for the particular cargo to a space based on the physical dimensions and weight of the

cargo. It does this by checking each arc along a path for feasibility until the entire path is built. Cargo cannot be placed in a hold if it conflicts with the Coast Guard Class of material already stowed in the compartment. Like cargo tends to get stored together as a result of these restrictions. This is why manually "seeding" initial cargo to compartments can significantly influence the results. The algorithm for placing cargo in the "best" hold is a combination of "stow penalties" and the anticipated time to off-load the cargo from a given compartment. This time is computed by adding up the times involved to transverse each arc of a path from the compartment to off the ship. Each placement decision depends on what cargo is being placed and on what cargo has already been placed in cargo holds, due to compatibility constraints. The routine is a "greedy" one in that it uses only the immediate state of the ship load to place the next piece of cargo and does not look at global follow-on consequences. Once all prioritized cargo has been placed, the process is repeated for the non-prioritized cargo. As can be imagined, the data requirements for this algorithm are extensive. The planner must provide priority numbers for every piece of equipment that must be placed first. (Current practice is to use these numbers primarily for vehicles.) The planner must also provide stowage penalties for the various compartments. Last and perhaps hardest of all, he must provide his time estimate for each

arc of every possible path for off-load from every compartment on the ship.

The proration process, though not an optimization, does have several advantages for the user and can serve as the basis for an alternate approach. The user is saved from making feasibility mistakes, which, considering the amount of data involved, is useful in and of itself. The ability to assign priorities, compartment penalties and a value, such as off-load cycle time can be used to create a "good" although not an optimal load. The ability to check cargo that has been prorated manually for feasibility is also a great benefit.

It must be emphasized that the CAEMS is not a replacement for the judgment of load planning experts. The primary benefits are speed, consistency, graphic output, and automatic report generation. The individual planner must still make the final decisions on cargo placement. Even though errors such as height, weight, or compatibility constraint violations are flagged, the user is still free to ignore the warning message. Once cargo placement is complete, there is no mechanism for assessing the quality of the load. What is missing is a way to compare the results of one load, either manually or automatically prorated, with an alternate feasible load. Several possibilities were considered from optimization literature.

IV. MEASURE OF EFFECTIVENESS FOR LOAD PLANNING

A. THE LOADING PROBLEM

The difficulty of this problem is that there is no single correct measure of what constitutes a "good" load much less an optimal one. On the one hand, if a ship is lightly loaded with extensive free space, every piece of cargo is easily reached and can be off-loaded at the appropriate time with no difficulty. This clearly supports the requirements of having equipment and supplies delivered to the beach when needed. On the other hand, amphibious ships are at a premium, and typically, even fully loaded the number of ships available for an amphibious operation is insufficient to carry all the desired equipment. This means that lightly loading only exacerbates this problem and essential equipment and supplies are left on the pier. What is the trade-off of essential equipment for other equipment, or for flexibility in the form of deck space? Unfortunately, this question cannot be answered in the abstract.

The bulk of available information of what is a "good" load is based on lessons learned after particular operations. If it worked, it was a good load. There are no data available on how to make changes to improve the process. Corporate knowledge is in the hands of a few expert officers in the Marine Corps and Navy. Loading the

ship the same way as last time if things have not changed too much is the usual policy. Expert opinion, without clear cut rules for trade-offs, is the only source of information to use as the basis for an optimization model.

An added complication is that many loads are equivalent. Loading a vehicle on the upper vehicle storage might be just as good as lower vehicle storage in many cases. Cargo hold five for some ammunition is not inherently different from cargo hold four. As a result, the solution space could prove to degenerate in nature. Even if this problem could be overcome, for the LHA there are dozens of compartments and over 1000 pieces of cargo. As a result, the enumerations for this problem could be exponential in nature.

B. APPROACHES

The current manual solution is clearly an unsatisfactory approach. The planner must use paper cut-outs of cargo, cut to scale, placed on scaled drawings of the ship's various cargo compartments. By carefully placing these templates, for square foot planning, and keeping track of height restrictions, a load plan is developed. This approach is both tedious and error-prone. The planner must keep track of all constraints such as weight and height; he must accurately cut or draw templates and must record the results of his efforts in numerous reports. Checking for whether a certain cargo item can fit through all the accesses to reach

a particular compartment is largely a matter of "knowing" from experience what does and does not fit through hatches and doors. The first step in improving this situation is computerization of the routine tasks involved.

CAEMS takes just such an approach. The system automates the manual process. The templates and deck drawings are stored in the database for the ship and types of cargo. The database of cargo and compartments ensures consistent and error-free planning. The exact size and shape of each cargo item is readily available with weight information as well. The elimination of the tedium and reduction of human error in the templating process is a tremendous improvement. By using AutoCAD to ease the job of placement of cargo, several arrangements can be tried in quick succession. From the database, required reports can be generated with little effort. Load plans can be saved on magnetic media to be used again in the future or shared with other planners. As previously discussed, the prototype system also checks for constraint violations and will automatically prorate cargo to specific compartments, if desired. CAEMS does not, however, solve the problem of automatically creating "good" loads.

B. OPTIMIZATION MODELS

The manual system and CAEMS do not evaluate the quality of the loads constructed. Several possible models were looked at to see if an improved load could be generated by a

personal computer-based system. The particular problem is similar to a knapsack problem with some key differences. In the knapsack problem, items with a particular value are available to be loaded into the "knapsack." A constraint, such as total volume that can fit into the knapsack, cannot be exceeded. The objective is to maximize the value of items loaded into the knapsack without violating the constraint. In general, this problem can be solved by an enumeration of all possible combinations of items that do not violate the constraint. The problem is combinatorial in nature and the time required to solve it becomes prohibitive as the number of items grows. In the case of shipboard loading, there is not one knapsack but several "knapsacks," one for each individual compartment. Items that do not fit in one compartment might fit in another. In addition, there are multiple constraints of height, weight, length and width, which are different for each compartment. The number of items to be loaded can be in the hundreds or thousands with dozens of compartments to choose from. Constraints on what types of hazardous material can be stored with each other means that the allowable items in each compartment can change depending on what items have been previously loaded. As a result of these complications, the knapsack approach was rejected as being too complex.

The next approach considered was that of multiple objective programming as described by Yu [Ref. 5] and by

Szidarovsky, Gershon and Duckstein [Ref. 6]. In this model a hierarchy of objective functions are developed that capture the required features of a good solution. The goal is to maximize the highest level objective subject to the given constraints. Normally, it is not possible to maximize this objective and still obtain satisfactory values for the lower level objectives. If the problem solver is willing to settle for some value less than the maximum possible for objective one (say 95% of the optimal value) the process continues by making this 95% criterion on objective one a constraint and maximizing objective function two. This process continues until every objective function is satisfied to a desired level. An example of this approach is:

$$\begin{aligned} \max \quad & y_1 = f_1(x) = 6x_1 + 4x_2 \\ \max \quad & y_2 = f_2(x) = x_1 \\ \text{s.t.} \quad & g_1 = x_1 + x_2 \leq 100 \\ & g_2 = x_1 + x_2 \leq 150 \\ & x_1, x_2 \geq 0 \end{aligned}$$

The ideal point $y^* = (500, 75)$ but this point is not in the feasible region. By reducing the values of y_1 below 500, a feasible solution can be obtained with good values for both objectives. A further explanation is found in Yu [Ref. 5].

The critical requirements for this method are a clear set of objective functions that can be arranged in priority order and a clear idea of what percent of an optimal solution is satisfactory as one descends to the lower objective function levels. With regard to the amphibious loading problem, several difficulties arise. The first is that there is no clear objective to maximize. Cargo in this case is not a bulk commodity where maximizing the number of pounds, for instance, would work. One might use an objective function for each major category of cargo but there is no obvious correct segmentation of cargo and no clear-cut percentage criterion for each category. The user does not think about the problem in these terms and no data are available to make the above choices. In addition, one would have to have an objective function for ease of access, or free space as a surrogate, which is not necessarily dominated by categories of cargo. Because of the problems involved, multiple objective programming was also rejected.

A third approach, goal programming as described in Lee [Ref. 7], seemed to have more promise than the other methods but also had severe shortcomings. In goal programming a set of priority levels is developed for each of several goals. The concept is to minimize deviations from a set of goal constraints with the priority levels determining multiplicative factors to apply to deviations from a given goal. A goal constraint equation is needed for each goal

involved in the problem. An example of converting the following problem to goal programming is from Lee [Ref. 7]:

$$\begin{aligned} \max \quad & Z = \$80x_1 + \$40x_2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 40 \\ & x_1 \leq 24 \\ & x_2 \leq 30 \\ & x_1, x_2 \geq 0 \end{aligned}$$

$$\begin{aligned} \min \quad & Z = p_1(d_1^+ + d_2^+ + d_3^+) + p_2 d_4^- \\ \text{s.t.} \quad & x_1 + x_2 + d_1^- - d_1^- = 40 \\ & x_1 + d_2^- - d_2^+ = 24 \\ & x2 + d_3^- - d_3^+ = 30 \\ & \$80x_1 + \$40x_2 + d_4^- - d_4^+ = \$10,000 \end{aligned}$$

The objective function shows that the highest priority, p_1 is the minimization d_1^+.

For the loading problem, the goals would be to load as much as possible in each category so negative deviations would not be penalized. Again, the problem arises of how many categories of cargo are appropriate. In addition, the user must decide what an adequate quantity, the goal, is for each category and what the trade-off penalties should be. There is still no clear way to include access in this process.

C. SCORING ALGORITHM APPROACH

In light of the difficulties with the above models another approach is needed. Rather than attempt to develop an optimization model, a scoring algorithm is developed, based on CAEMS, which allows the user to improve his solutions over time. In order to be effective, a scoring algorithm must consider the scope of the data available and the usefulness to the user of the scores developed. In a qualitative score the actual numbers developed are not important but the ranking of one solution versus another is what matters. One must be careful not to convey false impressions with such a number by differentiating too finely solutions that are essentially equivalent. A score of three or four digits is not any more meaningful than a score with two significant digits and could lead to bad conclusions. On the input side, the task of creating the necessary data must not be too onerous on the user. A small number of categories for scoring that the user can readily assign is much easier to implement. In addition, if the user cannot differentiate the relative values of one case over another, there is no point in assigning different categories to the two cases. The idea is to use enough categories to differentiate scores for situations that are clearly different without using so many categories as to confuse the user. Results that report differences where no meaningful ones exist must be avoided. When the final scores are

created it is often useful for the end user to see raw scores in each area, normalized scores and one overall grand score. An overall score allows for a quick comparison of one solution to another while each individual score allows the user to see in which area the solutions differ in value. By reporting raw scores as well as normalized scores any problems in the normalizing process can be pinpointed and corrected. The approach for amphibious loading is:

1. Develop meaningful areas to be scored.
2. Assign a reasonable number of categories in each scoring area.
3. Determine values for each category in each area.
4. Determine normalizing values for each scoring area.
5. Compute an overall total score based on the above.

This scoring technique offers significant advantages over the other methods reviewed. It is relatively easy to implement on a micro computer. It does not overburden the user. Required data are readily available. The necessary comparisons are easily made. The technique, if implemented, will lead to better solutions to the problem over time.

V. PROPOSED APPROACH

A. MEASURE OF EFFECTIVENESS (MOE) REQUIREMENTS

The proposed MOE builds on many advantages of the CAEMS prototype by examining the results of a given load compared to the landing operation that must supported, rather than optimizing the proration process. The MOE incorporates the key features of a good combat load. The way that the TEO should, and does, think about the on-load process is in terms of supporting the landing plan as developed by the CLF Operations Officer. If equipment must be left behind, it is operations who must make the final decision on what equipment to leave. If a ship is packed so tightly that equipment cannot be off-loaded in the required order, it is the landing plan that cannot be executed properly. Again either the Operations Officer must alter the landing plan or the ships involved must be loaded differently.

The key to these decisions is always to think about the problem in reverse order. The ship is off-loaded in the opposite order of the way it is loaded but it is the off-load that must drive the problem. A MOE must incorporate the importance of equipment arriving on the beach in the prescribed order and at the specified in the landing plan. The two things to look at in this process are:

1. Was the needed equipment loaded in the first place?
2. If it was loaded, can it be accessed for off-load at the proper time?

Since what makes one load better than another is only answered with reference to a landing plan, the proposed algorithm produces a MOE by scoring a given load against cargo available for on-load and against a given landing plan. As a result, the MOE is a number that can be used to compare alternate load plans that support the landing. The method computes a score for the decisions of what equipment to leave on the pier, and how densely to load each compartment and the ship as a whole. The MOE measures not only what equipment is loaded where, but also how easily the required order of off-load can be achieved.

The data requirements for the proposed method are less extensive than that required for goal programming. A small number of priority categories is needed to provide penalties for failure to load equipment and cargo. These can be broken down into a priority for the first n1 items or units followed by a lower priority for the next n2 items etc. Penalties are assessed against the chosen landing plan. Implicit in the MOE is the trade-off between free space percentage early in the off-load, created by leaving equipment behind (CLOP), and the desire for maximum cargo. The penalties for lack of free space continuously decrease as the operation proceeds because cargo off-loaded earlier contributes to available free space later in the off-load.

It is anticipated that when 50 percent of the cargo and equipment has been off-loaded, flexibility to stage and rearrange items as necessary is such that all cargo can be easily off-loaded in the required order. At this point the penalty for lack of free space drops to zero.

B. SCORING ALGORITHM

There are three main parts to the proposed scoring algorithm. The first deals with cargo that is available for on-load to support the amphibious operation but is never loaded (cargo left on the pier (CLOP)). To assess the importance of a particular piece of cargo the TEO must assign priority categories to all the cargo. The number of categories available should be a large enough number so that real differences in the "value" of cargo can be identified but not so large as to assign a large number of unique priorities. As a result of these considerations, ten priority categories were chosen: P1 to P10, with P1 being the highest priority, i.e., "must load," to P10 being the lowest, i.e., "load if it fits." There are approximately 1200 different cargo items in the LHA test database. A typical number for smaller ships might be 600 or 700. The system automatically assigns P10 to items not identified by the TEO. This will ensure that every piece of cargo is assigned a priority. The TEO probably will assign numbers to about half the cargo. The remainder would default to P10. The first part of the scoring is computed by summing

up the penalties for each piece of cargo left behind. A normalizing value is used to equate disparate units of cargo. This requires the entry of several hundred values but it is anticipated that these normalizing values would be the same across ship classes and landing plans and would only need to be entered once in a master data base for the particular type of cargo. The equation for this part of the scoring is as follows:

$$\text{raw CLOP penalty} = \sum_i \sum_j P_i X_{ij} U_j$$

where:

- i is penalty category;
- j is cargo type;
- U_j is a normalizing factor for cargo j;
- P_i is the penalty value for category i; and
- X_{ij} is the number of units of cargo type j, in category i left on the pier.

The next part of the score considers where cargo has been loaded in the ship. This requires the assignment of penalty values to every compartment onboard that can be used for stowing cargo. The CCO, as the expert maintainer of the SLCP, would assign these values for the ship. These values should be a reflection of the ease of off-loading cargo from a given space given normal circumstances. Should unique situations arise, the penalty category for a given

compartment could be adjusted. Five penalties were chosen from C1, the "hardest compartment to off-load," to C5, the "easiest" to off-load. The reason five categories were chosen is that typically there are areas that are easy to get to on a ship and others that are extremely difficult to reach, but the differences among many spaces are quite small. Mentally breaking up a ship into "hard" and "easy," one can imagine a rough categorizing but a continuous scale is not realistic. By allowing five categories, the trap of a strictly binary choice is avoided but the user is not called upon to make arbitrarily fine judgment calls that are not realistic. The equation for this part of the score is:

$$\text{raw compartment penalty} = \sum_{k} \sum_{i} \sum_{j} C_{ik} X_{jk} U_j$$

where:

- i is penalty category;
- j is cargo type;
- k is the compartment;
- C_{ik} is the penalty value i for compartment k;
- X_{jk} is the number of units of cargo type j in compartment k; and
- U_j is the normalizing factor.

Again note that a normalizing constant is employed to account for the variations among units of cargo. When deciding upon the proper units to consider for assigning the

compartment penalties, several possibilities were considered. The major features that make a piece of cargo easy or hard to off-load are weight, length, width, and height. In placing cargo on an elevator or conveyer, or moving around and through accesses, the main features are square footage or footprint (the square) and cubic volume (the cube). To simplify the problem, the assumption was made that the critical feature in assigning a penalty for stowage location was square footage. The reason this assumption was made is that for many cargo items there is no stacking effect because the item is not stackable. Vehicles are a good example of this kind of cargo. In the case of cargo that can be stacked, the manipulation to and from a space still depends primarily on the square footage, because height and weight limitations can be accounted for in the penalty value itself. If height or weight precluded an item from being placed into a particular stowage area then the penalty would not apply. Thus the penalties are assigned for each unit of cargo placed in a given compartment, with square footage as the normalizing factor.

The first two parts of the scoring mechanism are static in nature, either something was loaded into a particular compartment or it was left on the pier. The third part is dynamic and can only be computed by comparing the given load to the landing plan. The idea is to assign an "ease of off-load" penalty based on the percentage of free space in a

given compartment when an item is off-loaded. This free space value is the value of usable space after the stowage factor is considered. This stowage factor is an adjustment of the space available in a compartment to take into account the inability to pack densely and the requirements to provide space for proper tiedown of cargo in a compartment. Typically only 75 to 80 percent of a compartment's square footage is available for cargo after the above considerations. It is this 75 to 80 percent value that will be used for this scoring algorithm as this is the "real" space that can be used. When 50 percent of this space becomes available the penalty drops to zero. The concept is that free space will act as a surrogate for flexibility. The more free space in a compartment, the easier it is to reach required cargo either directly because there are numerous aisles or indirectly by restaging other cargo in open areas to reach the desired item. The key here is again the square footage footprint of the item because this is what affects aisles and the creation of open areas in a compartment. If a needed cargo item does have other things stacked on top of it, free floor space to restack is the critical necessity. It is assumed that there were no weight violations during the on-load so this constraint does not play a major part in scoring the off-load. The normalizing factor for the penalty equation is square footage footprint, as in the static compartment penalty, for the same reasons. The

dynamic part of this score component comes by looking for items to "off-load" in the order called for in the landing plan. This is done by providing a table derived from the plan that has the following information:

1. The serial numbers that will be part of each scheduled wave in wave order (the serial number is already in the cargo table). This serial number ties a particular cargo item to when it will be off-loaded.
2. The serial numbers of all cargo that is to be part of unscheduled waves that will be needed early in the amphibious operation.
3. Serial numbers of any other critical equipment not part of the general off-load.

The penalties themselves will be decreasing as a compartment empties. To maintain simplicity a continuous scale of free space was not used, rather discrete values were chosen from zero percent free space to 50 percent free space. As previously mentioned, at the 50 percent level no further penalties are assessed. The interval chosen for these discrete values was five percent. This level has the advantage of capturing differences among early serials and late ones without the difficulty of assigning too many different values within a given compartment for serials in the same wave. For simplicity the percent free space prior to "off-loading" a given cargo is used in the computation. Separate tables of cargo, serial numbers, compartment free space and landing plan information are maintained so that this process will not corrupt the actual load database. Each item of cargo is off-loaded in wave order as per the

landing plan and a score is given based on the state of the compartment from which it is taken. The free space percentage in that compartment is then updated by increasing the free space percentage based on the square footage made available by removing the item. This percentage is maintained continuously but free space penalties are based on every five percent increase as noted above. The process repeats for the next cargo item. Once all scheduled waves are off-loaded, unscheduled and other critical cargo serials are scored in the same manner. The process concludes when either all the scheduled, unscheduled, and critical serials have been scored or every compartment reaches a 50 percent free space value. The equation for this process appears below:

$$\text{raw free space penalty} = \sum_i \sum_j \sum_k F_{ik} X_{ijk} U_j$$

where:

- i is critical serials;
- j is cargo type;
- k is compartment;
- F_{ik} is the penalty value for serial i in compartment k;
- X_{ijk} is the number of units of cargo type j off-loaded in serial i from compartment k; and
- U_j is the normalizing factor.

Once each of these individual parts to the score has been computed it remains to combine them in a reasonable fashion so that a decision maker can make use of the information in determining the quality of the load. The actual scores depend on system penalty rates developed by the user and on the penalty categories assigned. Generally, the person who assigns the priority categories to each available cargo item will be the TEO. The person who assigns penalties for specific compartments will be the CCO. As a result, even with good, consistent penalty values the score for the material left on the pier may not be directly comparable to the score for loading cargo in particular compartments. In addition, the free space score may not be numerically comparable to either of the other scores because it is a measure that depends primarily on order of off-load. For these reasons it was felt that all three scores should be reported in the output for this algorithm. Weighting factors are used for each score to normalize them. The overall score is obtained by adding these normalized scores. The weights can be adjusted either to allow for differences of scaling among the individual scores or to place additional emphasis on one part of the score over another. The formula for the overall score is:

$$\text{Overall score} = W_1 \times (\text{CLOP penalty}) + W_2 \times (\text{compartment penalty}) + W_3 \times (\text{freespace penalty})$$

where W_1, W_2 and W_3 are weighting factors for the component scores.

The key considerations in developing the above scoring method were simplicity for the users responsible for assigning categories and a desire to capture an appropriate level of detail in the description of a given load that is provided by these categories. The decision to score penalty points was made to allow for flexibility in assigning scoring rates. The actual values derived are not important and can be scaled as noted above. The main issue is to provide a tool for comparison. By providing three separate scores and an overall score, each aspect of a load can be compared and the combined effects looked at as well. Because this system is computerized, it only takes a matter of minutes or perhaps hours to examine critical "what-if" scenarios. For instance, if the landing plan should change, affecting the order of off-load and therefore the free space penalty, how much worse or better is it? If it is determined that an item of CLOP must be loaded, what are the consequences of either a tighter load, affecting the free space penalty again, or perhaps leaving other cargo behind, affecting the CLOP penalty, the compartment penalty and the free space penalty? It is the ability to run these types of problems through the system quickly and produce numbers that can be meaningfully compared that is the true value of this computer-based method for determining a MOE.

VI. IMPLEMENTATION/VALIDATION

A. IMPLEMENTATION

To implement this model, the priority levels assigned by the user must be related to actual numeric values in the penalty equations. The usefulness of the resultant scores depends on these numbers being consistent with the trade-offs involved. For the penalty values for priority categories P1 to P10, the key is to relate the importance of each cargo type. For instance, if most vehicles are priority P1, how much more important are they than other types of cargo rated P2? If the tanks are twice as valuable, then the P1 penalty should be twice as high as the P2 penalty. One way to develop reasonable numbers then, is to have a user, or group of users, assign priority categories to every piece of cargo in a test load. The next step is to ask a series of comparison questions between the values of items in a given category. In this way the relative penalty numbers can be obtained. Note that the user is not asked to rate the relative importance in pairs of the 1200 or so items that make up a given cargo list. He is asked only to compare a select subset of cargo items in each of the ten priority categories. This reduction in the number of required comparisons is a key feature of using a limited number of penalty values rather than attempting to

determine the "utility" of every individual type of cargo. While some sensitivity to subtle value differences may have been lost, the gains in model simplification and ease of developing values from user information more than compensate for this.

Once these relative values have been obtained, a table of P1 to P10 is built in the Paradox database. This table is referenced during the computation of the CLOP score. The reasonableness of scoring penalties is determined by the ability to score a "good" load with a lower penalty than a "bad" load. The penalty values can be scaled by a factor W1 as noted previously. This factor serves two useful purposes. It allows values that are easy to compute with, i.e., integers of reasonable size, to be used for the penalty values, while allowing the final value to be a number that is easy for the user to relate to, say a number from one to 100. The other purpose for this scaling factor is to relate the relative weights of the three individual scoring methods. The actual algorithm for developing the score is straightforward and is written in Microsoft C. It takes each piece of cargo in the cargo table that is marked as being in the CLOP compartment (i.e., not loaded after proration), multiplies the number of units by the penalty category for those units and sums the results. A single pass through the cargo table accomplishes this, so the penalty is computed very quickly. A multiplication by the

scaling factor W1 produces the final results, which are then returned to a Paradox table for output.

The second portion of the scoring penalty, the compartment penalty, is developed in a similar manner. Here the trade-offs that must be compared are somewhat simpler. The user must first categorize each compartment on the ship into C1 to C5. Once this has been accomplished the relative ease of off-loading from a C1 compartment versus a C2 should be determined from the user. If a particular C1 compartment is twice as hard to off-load as a particular C2 compartment, then the penalty for C1 should be twice that of C2, etc. By doing comparisons of several compartments in each category an average relative difficulty of off-loading each compartment can be obtained. From these relative values, penalty numbers that are easy to compute with, can be developed. These values are stored in the Paradox database along with the penalty values for P1 to P10. Once the penalty values have been developed, the actual computation is very similar to that for CLOP. The cargo table is processed in a straight pass. Each unit of cargo is multiplied by the penalty for the compartment it has been prorated to and the results summed. As mentioned previously, the units of cargo must be normalized on a square foot basis. These results are multiplied by a scaling factor W2. The penalty values developed for C1 to C5, unlike the values for P1 to P10, will be very ship

dependent. While the user on a particular ship should have no difficulty in dividing the ship into five categories, the relative trade-offs among these five categories depend on numerous things that are platform dependent. As a result, it probably will be necessary to develop compartment penalties for each ship that uses this scoring system. Since the CLOP penalties are related to cargo items that will be the same from ship to ship, the development of P1 to P10 should not need to be repeated.

The third part of the score, the free space penalty, is not developed from the information available in the CAEMS database alone. This penalty is tied to the particular off-load order of the cargo based on a landing plan. As a result, to develop free space penalties a sample landing plan must be used. A Parodox database table is developed from this landing plan. The table contains the actual order that serials will be off-loaded based on scheduled waves, unscheduled waves, and critical cargo. This table is processed in wave order and each serial is "off-loaded," a penalty is computed, and the results are summed as mentioned in the scoring chapter. Penalty values for this portion are refined by comparing a given load plan against alternate loads. By loading critical early cargo in compartments without leaving free space, and then loosely loading this same cargo for other runs and comparing score results appropriate penalty rates can be determined. These F1 to

F10 rates can still be picked for ease of computation and need not be a straight ten to one type sequence. For instance, F1 could get a penalty of 20 and F2 one of 15. These type of values make intuitive sense because the flexibility of rearranging cargo should increase in a non-linear manner as more free space becomes available. Once values have been obtained for free space penalties, they are stored in the Paradox Penalty table with the other values. The scaling factor W3 is used in the same manner as W1 and W2. After values have been obtained for one landing plan, alternate landing plans against the same load plan can be used to check for consistency.

B. VALIDATION

All of the penalty developments mentioned above rely heavily upon the user. To validate the scoring system once these values are obtained, a panel of experts could be employed. Instructors and students training facilities such as Landing Force Training Command, Pacific could develop various mission scenarios. These scenarios would lead to load plans and landing plans that could be entered into the CAEMS system. From there, the scoring algorithm could be applied, keeping in mind that the C1 to C5 values must be derived for each ship. When several scores have been obtained for various missions, loads, and landing plans, these could be compared to each other. If the scoring system fails to differentiate between "good" and "bad"

loads, the individual scores could be studied to see where the inconsistencies are created. For instance, the CLOP score might be unreasonable because too high a value is placed on a particular type of cargo, or because the relative value of P1 in relation to P2 is too large. When these inconsistencies are discovered, the table of penalties can be adjusted in the database and the scoring redone against these same missions, load plans, and landing plans. In this fashion the scores can be refined until consistent results are obtained. To provide the maximum amount of information during this process, the output from the scoring algorithm is the raw score in each of the three categories, the three scores with weighting factors applied, and the total score for the load. In this way problems can be isolated to raw score, weighting factors, or total score. The key assumptions in creating the total score number are that the individual parts are independent and that a linear combination is appropriate. The individual parts are not entirely independent, however. The amount loaded does depend to some extent on the priority. How tightly a compartment is packed depends on the ease of off-loading the compartment. These interactions, though, are considered minimal and the simplicity of a linear additive model desirable. After many scores are developed, total scores and individual scores can be compared to confirm the reasonableness of these assumptions.

VII. CONCLUSIONS AND RECOMMENDATIONS

A critical factor in the success of an amphibious operation is how well the load plan supports the landing plan. The load plan must be driven by combat loading. The current situation of manual plan development based on a local store of previous plans is inadequate. A computerized approach is the key to solving this problem. The CAEMS prototype provides an easy-to-use, accurate database structure for the user. The error-checking features of the prototype ensure that all planned loads are feasible. Real-time development of alternate load plans in hours, instead of days, is now possible.

The scoring algorithm, using the CAEMS database, provides the ability to differentiate qualitatively among loads by computing penalty scores for the critical areas of CLOP, compartment location, and compartment access. This algorithm is implemented easily with available software. The requirements on the user are kept to a minimum with only ten priority categories, and five compartment categories. The resulting MOE is readily understood and can, over time, improve load planning through users learning what constitutes a better load. Plans can be readily stored and shared electronically to facilitate this process.

Further research is needed to develop "typical" missions and landing plans for a variety of situations. The algorithm needs to be run against these missions for particular ship classes. The results of these runs could be used to develop improved penalty rates for the priority categories and the compartment stowage categories. By running various missions, the effects of the free space penalty rates could be studied and adjusted as well. A training command such as Landing Force Training Command, Pacific could provide the basis for a team of experts to improve the scoring algorithm. Because the MOE is computed by a scoring algorithm, through the use of database tables revisions can be accomplished quickly. Various combinations of penalties and normalizing factors could be developed for each ship class and could even be adjusted for particular mission profiles.

The MOE produced by this scoring algorithm is cost effective, easy to implement, easy to use, and if fully developed and adopted will lead to improved loading of amphibious ships.

APPENDIX A

PAL CODE AND SAMPLE TABLES

DISCLAIMER: The reader is cautioned that the computer code provided in this appendix was developed for research purposes only. It is not of commercial quality and has not been exercised for all cases of interest. The author assumes no responsibility for possible damage to the CAEMS database by the use of this software. It is strongly recommended that the user back-up the database prior to experimenting with this code.

The code that follows was written in Paradox Application Language (PAL) using standard commands, to compute the penalty scores discussed in this research. The code consists of three types of Paradox scripts:

1. Queries that produce a subset of data tables.
2. Recorded menu selections and keystrokes.
3. Procedure scripts programmed with PAL commands.

Following the code section are examples of temporary tables created by the code and a sample output table for the algorithm. The run that produced these results was on a partially serialized cargo list and landing plan and are provided for illustration only.

```
Query

 Cargo | Cargo Identifier or TCN |
       | Check                   |
       |                         |
       |                         |

 Cargo | Area sf |
       | Check   |
       |         |
       |         |

 Cargo | Landing Serial | Parent Cargo Unit | Compartment |
       | Check          | blank             | Check       |
       |                |                   |             |
       |                |                   |             |

 Cargo |
       |
       |
       |
       |

Endquery
```

clopp

Query

Cargo	Unit Check	Cargo Identifier or TCN Check	Special Stow Class Check
Cargo	Template Label Check		
Cargo	Height in Check	Parent Cargo Identifier BLANK	
Cargo	Compartment "!CLOP"	Zone Identifier "!NONE"	

Endquery

ckioq1

```
Query

 Cargot1 | Cargo Identifier or TCN | Area sf  |  Compartment  |
         | Check                   | Check    | Check "!CLOP" |
         |                         |          |               |
         |                         |          |               |

 Cargot1 | Priority Rate |
         | Check         |
         |               |
         |               |

Endquery
```

serials

```
Query

 Cargot1 | Cargo Identifier or TCN | Area sf  | Landing Serial  |
         | Check                   | Check    | Check not blank |
         |                         |          |                 |
         |                         |          |                 |

 Cargot1 | Compartment |
         | Check       |
         |             |
         |             |

Endquery
```

compinfo

Query

Compart	Compartment Check	Total Area sf Check	Stowable Area sf Check

Compart	Remaining Area Unprorated Check

Endquery

thesis1

```
;*************************************************************************
;*Author Joseph Schneider, Machine 80386, Language PAL 3.0, March 1990    *
;*************************************************************************
;*This script queries the cargo table from CAEMS using the "cargoinf"     *
;*query script.  The answer table is renamed and the priority category    *
;*field is added.  The user must edit this field with priority values for *
;*cargo.  A blank will be interpreted during computations as the lowest   *
;*valued priority(P10).                                                   *
;*************************************************************************

proc thesis1()
Play "cargoinf" Do It! ;query cargo table for needed fields
clearall               ;clear the workspace
rename "answer" "cargot1";save the query table
play "cargomod";  modify priority field
endproc
thesis1()
```

thesis2

```
;*************************************************************************
;*Author Joseph Schneider, Machine 80386, Language PAL 3.0, March 1990    *
;*************************************************************************
;*This script queries the "cargot1" table from thesis1 using the "serials"*
;*query script.  The answer table is renamed and sorted by serial number. *
;*************************************************************************

proc thesis2()
Play "serials" Do_It! ;get cargo that has serials assigned
 Clearall
Rename "answer" "cargot2" ;preserve answer in temporary file
view "cargot2"            ;place file in work space
sort "cargot2" on "Landing Serial" ; sort by serial number
endproc
thesis2()
```

thesis3

```
;*************************************************************************
;*Author Joseph Schneider, Machine 80386, Language PAL 3.0, March 1990    *
;*************************************************************************
;*This script queries the cargo table from CAEMS using the "compinfo"     *
;*query script.  The answer table is renamed and a compartment penalty    *
;*field is added by the script "compmod".  The user must fill in these    *
;*values prior to playing the compute script.                             *
;*************************************************************************

proc thesis3()
Play "compinfo" Do It! ;get relevent compartment info
Clearall
Rename "answer" "compt1" ;save the answer in new table
play "compmod"; add compartment penalty field
View "compt1"
endproc
thesis3()
```

thesis4

```
;*************************************************************************
;*Author Joseph Schneider, Machine 80386, Language PAL 3.0, March 1990    *
;*************************************************************************
;*This script queries the "cargot1" table using the "clopq1"              *
;*query script to find all of the CLOP cargo.                             *
;*************************************************************************

proc thesis4()
Play "clopq1" Do_It!    ;find all clop records
Rename "answer" "clopt1"
endproc
thesis4()
```

maketemp

```
;*************************************************************************
;*Author Joseph Schneider, Machine 80386, Language PAL 3.0, March 1990    *
;*************************************************************************
;* This script is a driver script to produce the needed temporary tables.  *
;* The tables produced are:                                                *
;*  "cargot1"    a table produced by a query of the CAEMS cargo table      *
;*  "cargot2"    a table of serialized cargo by a query of "cargot1"       *
;*  "compt1"     a table produced by a query of the CAEMS compartment table*
;*************************************************************************
play "thesis1"
play "thesis2"
play "thesis3"
play "thesis4"
```

cargomod

```
;*************************************************************************
;*Author Joseph Schneider, Machine 80386, Language PAL 3.0, March 1990    *
;*************************************************************************
;*This script is a record script that modifies the temporary table        *
;*"cargot1" and provides for range checking when inputing penalty         *
;*categories                                                              *
;*************************************************************************

{Modify} {Restructure} {cargot1} Down Down Down Down
 "Priority Rate" Tab  "A2" Do_It!
Menu {Modify}
{Edit} {cargot1} Menu {ValCheck} {Define} Right Right Right
Right Right Enter {LowValue} {1} Menu {ValCheck} {Define} Enter
{HighValue} {10} Menu {ValCheck} {Define} Enter {Picture} {{#}#}
Menu {ValCheck} {Define} Enter {Default} {10} Menu Esc Do_It!
Menu {Scripts} {End-Record}
```

```
;*****************************************************************************
;*Author Joseph Schneider, Machine 80386, Language PAL 3.0, March 1990      *
;*****************************************************************************
;*This script is a record script that modifies the temporary table          *
;*"compt1" and provides for range checking when inputing penalty            *
;*categories.                                                               *
;*****************************************************************************

{Modify} {Restructure} {compt1} Down Down Down Down
 "Compartment Penalty" Tab   "s"  Do_It!
Menu {Modify} {Edit} {compt1} Menu
{ValCheck} {Define} Right Right Right Right Right Enter {LowValue}
{1} Menu {ValCheck} {Define} Enter {HighValue} {5} Menu {ValCheck}
{Define} Enter {Default} {5} Do_It! Menu {Scripts} {End-Record}
```

```
;*************************************************************************
;*Author:Joseph Schneider, Machine 80386, Language PAL 3.0,    March 1990    *
                    *
;*************************************************************************
;*The compute script consists of three procedure: computeclop(),           *
;*computecomp() and computefree().  These procedures use the temporary      *
;*tables created by Maketemp which must be run first.  Together they        *
;*compute and store the scoring penalty values into the table "outputt1".   *
;*The input tables used are:                                                *
;*                                                                          *
;*   "cloprate", "comprate", "freerate" and "weights"  for penalty rates and *
;*weighting factors,                                                        *
;*                                                                          *
;*    "cargot1", "cargot2","compt1" and "landplan" for information about    *
;*cargo location, cargo serialization, compartment information and serials  *
;*in the landing plan.                                                      *
;*************************************************************************

rawclop=0
view "cloprate"
copytoarray rates ;clop penalty rates
view "comprate"
copytoarray crates ;compartment penalty rates
view  "freerate"
copytoarray frates ;freespace penalty rates
view "weights"
copytoarray w     ;weighting factors for scores
clear
clearall
proc computeclop()
view "clopt1"                ;cargo left on pier temporary table
scan "clopt1"
If not isblank([clopt1->priority rate])       ;no priority = P10
then rawclop = rawclop +  rates[numval([clopt1->priority rate]) + 1 ]
          * [clopt1->Area sf]
else rawclop = rawclop + rates[11] * [clopt1->Area sf]
endif
@10,10 ?? "Raw Clop Penalty is " + strval(rawclop)
endscan
edit "outputt1"                              ;store penalty values
[outputt1->Raw CLOP Penalty] = rawclop
[outputt1->Weighted CLOP Penalty] = w[2] * rawclop
Do_it!
endproc
```

```
proc computecomp()                              ;compute compartment penalties
clear
clearall
view "compt1"                                   ;compartment temporary table
view "cargot1"                                  ;cargo temporary table
rawcomp = 0
for i from 1 to nrecords("compt1")
   compmatch = [compt1->Compartment]            ;get compartment to match
   if compmatch <> "!CLOP"                      ;no compartment penalty for CLOP
      then penalty = crates[[compt1->Compartment Penalty] +1]
           scan  for  compmatch = [cargot1->Compartment]
               rawcomp = rawcomp + penalty * [cargot1->Area sf]
               @10,10 ??"Raw Compartment Penalty is " + strval(rawcomp)


           endscan
   endif
   upimage                                      ;move view to compt1
   down                                         ;move to next record
   downimage                                    ;move view back to cargot1
endfor                                          ;end of compt1 records
 @10,10 ??"Raw Compartment Penalty is " + strval(rawcomp)
edit "outputt1"                                 ;store raw and smooth scores
[outputt1->Raw Compartment Penalty] = rawcomp
[outputt1->Weighted Compart. Penalty] = w[3] * rawcomp
Do_it!
endproc
```

```
proc compfreespace()                          ;compute free space and total score
clear
clearall
view "compt1"                                 ;temporary compartment table
view "cargot2"                                ;temporary serialized cargo table
view "landplan"                               ;landing plan serial table
rawfree = 0
for i from 1 to nrecords("landplan")          ;for each serial in landing plan
   serialmatch = [landplan->serial number]    ;get serial number to match
   upimage                                    ;move view to cargot2
   scan for serialmatch = [cargot2->landing serial]  ;find serial compartment
        compmatch = [cargot2->compartment]           ;in cargot2 table
        areacargo = [cargot2->area sf]               ;get area of cargo
   upimage                                    ;move view to compt1
   scan for [compt1->Compartment] = compmatch ;find match and compute %free
        percentfree =
              [compt1->Remaining Area Unprorated]/[compt1->Stowable Area sf]
        if percentfree < 0.525                      ;in the less than 50% bin
           then
               f = round(percentfree * 20,0) + 1    ;+ 1 is for table name
               if f = 1 then f = 2                  ; if in 0% bin need to
               endif                                ; add 1 to get past table
                                                    ;name
               penalty = frates[f]                  ;get penalty rate from table
               rawfree = rawfree + penalty * areacargo ;add new penalty
               edit "compt1"                        ;update area "offloaded"
               [compt1->Remaining Area Unprorated] =
                           [compt1->Remaining Area Unprorated] + areacargo
               Do_it!
        @10,10 ??"FREESPACE PENALTY " + strval(rawfree)
        endif                                  ;compartment free space updated
    endscan
    downimage                                  ;move view back to cargot2
endscan                                        ;end scan for cargo in serial
downimage                                      ;move view back to land plan
    down                                       ;next record in land plan
endfor                                         ;finished with land plan
edit "outputt1"                                ;store free space and overall
[outputt1->Raw Free Space Penalty] = rawfree   ;scores
[outputt1->Weighted F. Space Penalty] = w[4] * rawfree
[outputt1->Overall score] = [outputt1->Weighted CLOP Penalty] +
                            [outputt1->Weighted Compart. Penalty] +
                            [outputt1->Weighted F. Space Penalty]
Do_it!

Clearall                                       ;clear the workspace
endproc
computeclop()                                  ;call procedures to do
computecomp()                                  ;the computations and
compfreespace()                                ;store the results
```

Temporary Compartment Table

Compartment	Total Area sf	Stowable Area sf	Remaining Area Unprorated	Compartment Penalty
ICLOP	2	2	2	5
AVIATION ARMORY	534	534	534	5
FAE ORDNANCE	534	534	534	5
FLIGHT DECK	94422	71485	71485	3
HANGAR DECK	20932	19154	339	5
LANDING FORCE WEAPONS	2062	2062	2062	2
LCAC	2062	1791	1791	2
LCM 6	450	448	448	1
LCM 8	633	633	633	5
LCM 8 (ALUM)	724	724	724	2
LCPL	68	58	58	5
LCU 1466	1771	1748	10	5
LCU 1610	2247	2164	2164	3
LCU 1610-2	2247	2164	2164	5
LCU 1610-3	2247	2164	2164	2
LCVP	137	131	131	2
LOWER 4 (PORT)	1296	1296	695	2
LOWER 4 (STARBOARD)	1468	1468	1	2
LOWER 5 (PORT)	891	891	374	3
LOWER 5 (STARBOARD 1)	365	365	365	3
LOWER 5 (STARBOARD 2)	421	421	421	3
LOWER 5 (STARBOARD 3)	406	406	143	3
LOWER 5 (STARBOARD 4)	171	171	86	1
LOWER 9	709	709	709	5
LOWER VEHICLE STOWAGE,AFT	11749	9212	9212	2
LOWER VEHICLE STOWAGE,FOR	5040	4138	4138	5
POL	1323	1323	1323	5
PYROTECHNIC LOCKER	299	299	243	5
SHIP'S ARMORY	1297	1297	1297	5
SHIP'S TANKS (BULK POL)	1297	1297	1297	5
SPECIAL STOWAGE	1297	1297	1297	2
TROOP ARMORY	1297	1297	1297	2
TROOP SPACE	20000	20000	19575	2
UPPER 4 (PORT)	1297	1297	1297	2
UPPER 4 (STARBOARD)	1441	1441	1441	4
UPPER 5 (PORT)	1211	1211	8	4
UPPER 5 (STARBOARD)	1358	1358	292	2
UPPER 9 (PORT 1)	754	754	108	1
UPPER 9 (PORT 2)	687	687	670	2
UPPER 9 (STARBOARD)	1536	1536	1536	3
UPPER VEHICLE STOWAGE	20595	17961	13305	5
WELL DECK	25298	17173	7695	4
WP ORDNANCE	495	495	1	3

Temporary Cargo Table

Cargo Identifier or TCN	Area sf	Landing Serial	Compartment	Priority Category
!68	107		!CLOP	10
!69	107		!CLOP	
!70	107		!CLOP	
!77	107		!CLOP	
!81	107		!CLOP	
!82	107		!CLOP	
!83	107		!CLOP	
!93	107		!CLOP	
0000	4		TROOP SPACE	
0001	4		!CLOP	
0001	25		!CLOP	
0001-1	4		TROOP SPACE	
0002	5		!CLOP	
0002-1	4		TROOP SPACE	
0003	1		!CLOP	
0003-1	4		TROOP SPACE	
0004	4		TROOP SPACE	
0005	4		TROOP SPACE	
0006	4		TROOP SPACE	
0007	2	0351	!CLOP	
0007	4		TROOP SPACE	
0008	3	0351	!CLOP	
0008	4		TROOP SPACE	
0009	4		TROOP SPACE	
0009	76		!CLOP	
0010	4		TROOP SPACE	
0011	95		!CLOP	
0011-1	4		TROOP SPACE	
0012	3		!CLOP	
0012	4		TROOP SPACE	
0013	4		TROOP SPACE	
0013	135	1745	!CLOP	
0014	4		TROOP SPACE	
0015	4		TROOP SPACE	
0016	4		!CLOP	
0017	5		!CLOP	
0018	10		!CLOP	
0018-1	4		TROOP SPACE	
0019	13		!CLOP	
0019-1	4		TROOP SPACE	
0020	14		!CLOP	
0021	14		!CLOP	
0022	2		TROOP SPACE	
0024	4		TROOP SPACE	

CLOP Penalty Rates

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
50	45	40	35	30	25	20	15	10	5

Compartment Penalty Rates

C1	C2	C3	C4	C5
50	45	35	25	15

Freespace Penalty Rates

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
100	95	85	75	60	45	25	15	10	5

Outputs Values from Scoring Algorithm

3/23/90

Overall Score	Weighted CLOP Penalty	Raw CLOP Penalty	Weighted Compartment Penalty	Raw Compartment Penalty	Raw Free Space Penalty	Weighted Free Space Penalty
272921	39358	98395	232867	582169	3480	696

APPENDIX B

GLOSSARY OF TERMS

Some definitions for terms used throughout this thesis are provided below to help in discussing the on-load/ off-load planning process.

Administrative loading	A method of loading that ensures the maximum amount of cargo is loaded.
CAD	Computer-aided design. A software package that assists the user in designing various layouts. In this application, AutoCAD provides templating and cargo placement utilities.
Cargo placement	The locating of cargo in a particular spot within a compartment. Performed by CAEMS by the use of AutoCAD routines.
CATF	Commander, Amphibious Task Force, a senior naval officer responsible for all aspects of the amphibious operation.
CCO	Combat Cargo Officer, responsible to the commanding officer of an individual ship for all aspects of the on-load and off-load of men and equipment.
CLF	Commander, Landing Force, a senior marine in charge of the assault on the beach.
CLOP	Cargo left on the pier. Those items that are left behind after the on-load is complete.
Coast Guard Class	Certain explosive materials must be stowed separately from other materials. Each of these restricted materials are given a class designation. Rules for which classes can be stored with which other classes have been devised and must be observed.

Combat loading	A method of loading a ship that ensures an off-load in the order that equipment is needed on the beach.
Cycle Time	The time that the planner may assign for going along a particular path to off-load cargo from the ship.
Embarked unit	A military organization that comes aboard a ship as a distinct entity.
Landing plan	The plan for landing on the beach. It contains all of the details of when units and equipment will be sent to the beach.
On-load	The process of loading cargo onto the ship. The routes taken to load are not necessarily the same as those taken to off-load. In particular, the LHA uses ramps at the side of the ship for on-load but in an actual amphibious operation the well deck and flight deck are the locations for off-load.
Off-load	The process of taking cargo off of the ship.
Priority Cargo	The user can assign a priority number to cargo as he desires. In automatic proration cargo is assigned to compartments in priority order then non-prioritized cargo is prorated.
Proration	The process of manually or automatically allocating cargo to compartments on a ship.
"Seeding" cargo	The process of manually allocating some cargo to various compartments to enhance the performance of the automatic proration routine.
Serial number	A serial number is an identifying number that is unique to a particular unit or cargo. The landing plan details the order in which serials are sent ashore.
Stow Penalty	A penalty value that the user may assign to a compartment making it a less desirable location for cargo placement.

TEO	The Team Embarkation Officer. He is responsible to the CLF for all aspects of the on-load.

LIST OF REFERENCES

1. Computer Aided Embarkation Management System Prototype (CAEMS-1), User's/Operations Manual, Stanley Associates, August 1989.

2. Computer Aided Embarkation Management System Prototype (CAEMS-1), Training Manual, Stanley Associates, August 1989.

3. "Ground Combat Operations, OH6-1," p. 6-1 to 6-21, Commanding General, Marine Corps Combat Development Command, January 1988.

4. Joint Chiefs of Staff Publication 1.

5. Yu, Po-Lung, Multiple-Criteria Decision Making, pp. 67-91, Plenum Press, 1985.

6. Szidarovsky, F., Gershon, M.E., and Duckstein, L., Techniques for Multiobjective Decision Making in Systems Management, pp. 22-23, Elsevier Science Publishers, B.V., 1986.

7. Lee, Sang M., Goal Programming For Decision Analysis, pp. 39-43, Auerbach Publishers, Inc., 1972.

8. Embarkation Planner's Handbook, Landing Force Training Command, Pacific, December 1988.

BIBLIOGRAPHY

"The Detailed Landing Plan," Student Handout, OPS026, Landing Force Training Command, Pacific, December 1989.

"Guide to the Paradox Personal Programmer," release 3.0, Borland International, 1988. (Parodox is a registered trademark of Borland International.)

Hallefjord, Ä. and Jörnsten, K., "A Critical Comment On Integer Goal Programming", Journal Operational Research Society, Volume 39, Number 1, 1988.

Chvátal, V., "Hard Knapsack Problem," Operations Research, Volume 28, Number 6, November-December 1980.

Kelly-Bootle, S., Mastering Quick C, Sybex Inc., 1989.

Microsoft Quick C (Version 2.0), Toolkit, Microsoft Corporation, 1988. (Quick C is a registered trademark of Microsoft Corporation.)

Omura, G., Mastering AutoCAD, 3rd edition, Sybex Inc., 1989. (AutoCAD is a registered trademark of Autodesk, Incorporated.)

Paradox User's Guide, release 3.0, Borland International, 1988. (Parodox is a registered trademark of Borland International.)

Paradox PAL User's Guide, release 3.0, Borland International, 1988. (Parodox is a registered trademark of Borland International.)

Rasmussen, L. M., "Zero-One Programming With Multiple Criteria," European Journal of Operation Research, Volume 26, p. 83, 1986.

Ship to Shore Movement, Self-Paced Instruction, EMB026, Landing Force Training Command, Pacific, November 1987.

Simpson, A., Mastering Parodox, 4th edition, Sybex Inc., 1989.

Steuer, R.E. and Choo, E.-U., "An Interactive Weighted Tchevycheff Procedure for Multiple Objective Programming," Mathematical Programming, Volume 26, Number 3, August 1983.

White, D.J., Optimality and Efficiency, John Wiley and Sons, Ltd., 1982.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5000	2
3.	Chief of Naval Operations (OP 04P) Navy Department Washington, D.C. 20350	1
4.	Chief of Naval Operations (OP 813B) Navy Department Washington, D.C. 20350	1
5.	Commandant of the Marine Corps Code LPC-4 Headquarters, U.S. Marine corps Washington, D.C. 20380-0001	2
6.	Commander, Amphibious Group 3 P.O. Box 201 San Diego, California 92136	2
7.	Commander, Amphibious Squadron 3 FPO San Francisco, California 96601-5806	2
8.	Professor Gordon H. Bradley, Code OR/Bz Department of Operations Research Naval Postgraduate School Monterey, California 93943-5000	4
9.	Lieutenant Commander William J. Walsh Code OR/Wa Department of Operations Research Naval Postgraduate School Monterey, California 93943-5000	2
10.	Lieutenant Commander Roger Stamp, Code OR/30 Department of Operations Research Naval Postgraduate School Monterey, California 93943-5000	1

11. Professor David A. Schrady, Code OR/So 1
Department of Operations Research
Naval Postgraduate School
Monterey, California 93943-5000

12. Professor Michael G. Sovereign, Code OR/Sm 1
Department of Operations Research
Naval Postgraduate School
Monterey, California 93943-5000

13. Lieutenant Commander Joseph M. Schneider 4
17606 Azucar Way
San Diego, California 92127